



## Compilando código

### Transcripción

[00:00] Ahora que tenemos ya nuestro ambiente preparado, para comenzar a crear nuestro primer archivo en Java, perfecto. Entonces no vamos a usar, por ahora, algún IDE o editor de código profesional, quiero que vayamos paso a paso. Para esto, entonces voy a enseñarles cómo es cada sintaxis como un bloc de notas simple.

[00:28] Vamos a abrir aquí. Recuerden que para escribir código necesitamos cualquier editor de texto, no necesita ser un editor profesional o especializado solo para código. Vamos solamente a ver cómo sería escribir un código en Java. Si venimos de otros lenguajes de programación o nosotros quisiéramos por ejemplo imprimir en la consola un "Hola Mundo", vamos a escribir aquí.

[01:00] Para esto, si queremos pasarlo a un lenguaje por computador, podríamos intentar algo como esto: `print("Hola Mundo")` y eso debería ser suficiente. En el caso de Java necesitamos un poco más de comandos para hacer entendible para la Java Virtual Machine, que es lo que nosotros queremos.

[01:26] En este caso la instrucción para imprimir alguna palabra en la consola sería esta de acá: `System.out`, porque queremos imprimir algo fuera, `print line`, `ln`. Y con eso nosotros conseguimos imprimir: "Hola Mundo" en la consola. Siempre en Java al final necesitamos poner un punto y coma que es digamos el delimitador, que la Máquina Virtual Java va a entender que aquí terminamos una instrucción. Perfecto.

[02:01] ¿Es esto suficiente para imprimir "Hola Mundo"? Aún no. Para que esta instrucción pueda ser ejecutada y compilada a un ByteCode necesitamos primero incluirla dentro del contexto de una clase. Aún no se preocupen por entender qué es una clase o algo así. Estamos yendo por partes. Vamos a poner esto en un contexto public class ejemplo.

[02:34] Y por contexto yo me refiero más o menos a esta sintaxis de aquí. Copiamos esto y lo pegamos aquí adentro. Entonces, con esto ya sabemos que existen System.out.println se está desarrollando dentro de ejemplo. Perfecto. ¿Esto es suficiente? Aún no. Porque para que ejemplo pueda imprimirse, necesita ser ejecutado por alguien. En este caso, estamos hablando de un método.

[03:17] Entonces, tiene que haber un método ahí que ejecute y le diga que quiero que imprimas "Hola Mundo" en la terminal, porque la clase es solamente un contexto que tenemos ahí. Entonces, existe un método en Java, en lenguaje Java, que es usado para imprimir instrucciones directamente en la terminal, siendo ejecutado directamente en el código, por ejemplo es el public static void main.

[03:56] No se preocupen aún por entender qué significa cada una de estas cosas, estamos solamente evaluando cómo sería hacer un "Hola Mundo" en el lenguaje Java. Este método va a tomar como argumentos o como parámetros un conjunto de strings. En este caso, si ustedes ven ejemplos en internet, pueden encontrar strings de esta manera o strings también de esta manera.

[04:30] No se preocupen aún en entender cuál es la diferencia, prácticamente no hay diferencia, es lo mismo. Y de la misma forma anterior incluimos este aquí dentro y regresamos aquí. Perfecto. Entonces ya tenemos nuestra clase ejemplo con el método Main, que él es el primero que va a ser ejecutado, y él va a mandar a imprimir en la consola "Hola Mundo".

[05:00] Entonces, vamos a probar, vamos a ver si está bien, vamos a guardar esto, vamos a guardar esto en nuestro escritorio con el nombre "Ejemplo.Java". Un dato curioso es que el nombre del archivo tiene que ser el mismo nombre de la clase para poder ser entendido por la máquina virtual de Java. Ya vamos a profundizar en estos conceptos más adelante, por ahora vamos solamente a hacer esto. Guardamos aquí.

[05:38] Tenemos Ejemplo.Java y aquí vamos a buscar nuestro archivo creado. Perfecto. Aquí encontramos a Ejemplo.Java. Si recordamos en el video anterior, para poder compilar y generar el ByteCode necesario, necesitábamos el comando JavaC. Como ya lo tenemos en el path instalado, vamos a ejecutar JavaC Ejemplo.Java a ver qué sale. Vamos ahí.

[06:18] En teoría no hizo nada, pero si vemos aquí, ya generó Ejemplo.class. Este es el archivo con el ByteCode que va a ser interpretado por nuestra máquina. Entonces vamos a ejecutar ese archivo aquí con el comando Java. Limpiamos un poco la pantalla. Y ahora, con el comando Java, ejemplo.

[06:46] Para ejecutar nosotros el método, no necesitamos ponerle extensión del archivo punto class por ejemplo, porque Java automáticamente lo que va a hacer es buscar el punto class internamente, porque él va a entender, por el comando Java, que ya tiene que haber un archivo compilado con ese nombre ejemplo. Ejecutamos, damos enter y tenemos ahí nuestro "Hola Mundo".

[07:12] Fue exitoso nuestro primer test, ya estamos escribiendo nuestro primer código en Java. Y ahora vamos a ver ciertas curiosidades aquí que podrían surgir en un día a día, por ejemplo, ¿qué pasaría si olvido colocar el punto y coma al final de "Hola Mundo"? Yo voy a guardar aquí. A ver. Vamos a guardar. Voy a proceder a eliminar este archivo punto class.

[07:45] Si yo voy a ejemplo, no se ha encontrado la clase principal, perfecto, está muy bien, entonces voy a compilar nuevamente mi archivo Ejemplo.Java,

pero sin el punto y coma. Al final, vamos a ver qué sale de ahí, de ejemplo, y listo, nos dice que esperaba un punto y coma al final de "Hola Mundo".

[08:09] Como ustedes pueden ver, el compilador de Java siempre nos da ciertas indicaciones, ciertas pistas de dónde podría estar el error que hemos cometido. Eso es lo que llamábamos error de compilación. Los errores que surgen cuando no podemos transformar código Java en ByteCode son errores de compilación.

[08:31] Entonces, agregamos aquí. ¿Y qué pasaría también si olvidamos poner el tipo de datos de argumentos? Vamos a guardar ahí y vamos a ejecutar el mismo comando. Recuerden que ya tenemos el punto y coma al final. ¿Qué dice? Esperaba un identificador aquí, y nos señala más o menos también dónde puede estar nuestro error.

[09:00] Entonces, como pueden ver, la máquina virtual de Java, entre el conjunto de utilidades y librerías extra que nos dan a nosotros como desarrolladores, en este caso JDK, nos facilita mucho la vida para encontrar errores de sintaxis, detectar por ejemplo editores de código, avanzados incluso, nos detectan dónde podríamos tener un error.

[09:26] Los editores mismos, si nosotros tenemos un error en este punto, nos van a subrayar en rojo diciendo: "por si acaso tienes un error ahí". Entonces, ya que conocemos lo básico para escribir código en Java y lo vemos en un simple bloc de notas, vamos a pasar ahora sí ya a tocar un entorno ya profesional de desarrollo de software, en este caso me estoy refiriendo a un IDE.

[09:54] Entonces, nos vemos en la siguiente clase para ya instalar un IDE y comenzar a escribir código de verdad. Nos vemos.